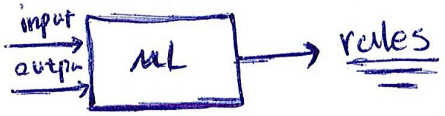
machine_learning:

- machine learning is branch of AI and computer science in which tries to learn and gain knowledge from input data to have great prediction ⇒ IBM definition ♡
- without explicit coding



Deep_learning:

IBM
- is subset of machine learning, which contains neural network
- why Deep? using three or four or more layer
- neural network is heartbreak
- computing system → tries to mimic human brain

Computer vision:
- Field of AI which gives ability to machines to process and exctract meanigful information from Images/videos like human.

Computer vision application:
1) Image classification
2) object detection
3) Pattern recognition → Face detection → Facial emotion ★ → ★ → get started
4) Face detection
⋮

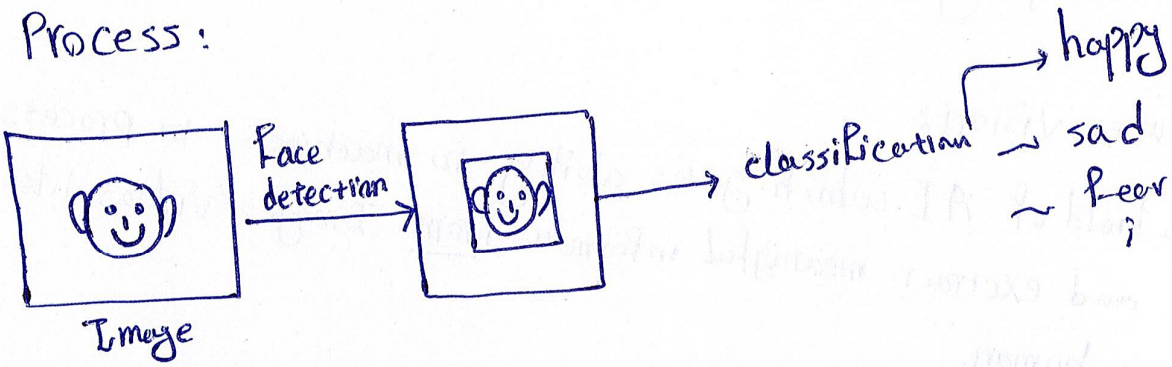
Facial emotion recognition = FER (Vietnamese)

explaintion: FER is computer vision task aimed at identifying and categoring emotional expression depicted on human face by analyzing Feature of face : eyebrows, eye, mouth,

mapping them to { anger, fear, suprise, saddnes }

- use Images / videos
- real times
- Facial emotion is form of non-verbal communication

Process:



application:

Education { - monitor student' attention
- detect a user's reaction to a topic
- help to design a affective educn system.
:

health care { - detect patient's feeling.
- observe patient condition.
}

crime detection => { spot shoplifter
:
}

- Preprocessing

1) grayscale Image)
- they help to have less computation
- time saving
- without having any significant effect on accuracy specially on Facial emotion recognition.

2) resizing)
- Images are in different Dimention ⇒ they have to convert to the same size → because we have introduced input layer.

3) Data augmentation)
- making more images based on feature which have been Provided with current images.

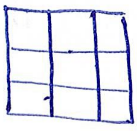
4) Face Detection:
- the most important part of preprocessing
- algorithms/structures do not need background
- Faces are more important.

4,1) Viola-jones algorithm:
- humans can detect many objects easily (cars - trees, ...)
- this activity sounds crazy when it comes to computer.
- How does viola jones work?

★ Viola-jones' goal is to detect Face.

- to detect Face:

1)- they look at the Frontal view of Face.

2) algorithms chunk image into section ⟹ 

3) examine each section and try to see if it sees any Features of Face.

4) if it doesnt see any Features ⟶ move to next section.

4.1) Features are ⟹ eys, mouth, nose, cheeks,...

★) Face can be considerd as map and Face Features are landmark on maps.

★) algorithm is like a person trying to find the area that has all the landmarks in one location.

★) every time the algorithm detects a Face it marks where it found it on the pictures and keeps repeating. Process until it has checked whole images.

★★★) when the algorithm has finished scanning the Face:

- there are bounch of boxes that mark where Fa located
- average on all location

Viola jones Face detection:

- Har Features
- integral images
- adaboost
- cascading.

- it was trained with Faces and non_Face data to predict unseen Face → real time - video - ..

- training Part { showing loads of imge and telling the Face
  ~ ~ ~ ~ ~ ~ is not Face

- goal → tries to train and gives ability to computer to understand what Face is and what is not Face by using picture data.

- once computer/Program is trained → it can extract Features certain Feat

training process: - all certain Feature will be stored in a File

testing part: 1) taking that important File

2) take new image in File

3) check all Features and apply it on image

4) if it passess all the feature comparision → Face

- How are we going to extract Features?

* First of all lets talk about how to extract edge

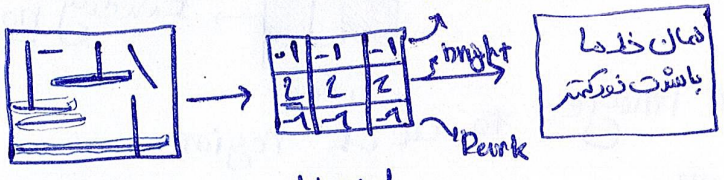


Image Kernel

single horizontal line detection

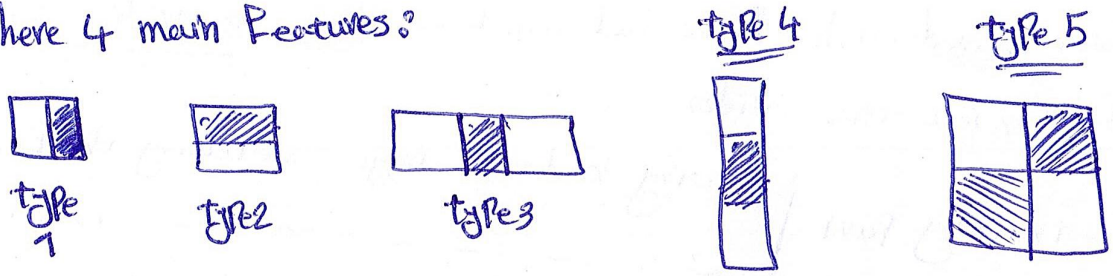
*) by using this kernel: Finding the single line (horizontal)

*) Kernel moves on image

*) gives us the greatest information about location horizontal line

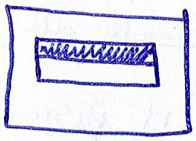
haar_Features:

- they are more similar to convolution kernels
  - ↳ try to find Feature and detect the presence of that Features.
- there 4 main Features:



- they are the same of kernel ⇒ visualize.

1
-  → black → 1, white → -1 → 
- when we apply this Feature to an image
  - ↳ Function: subtract the pixel values under white region from pixels from black region
    - ↳ output → single value

2
- consider another type of Features
- type2 ⇒ apply it to image →  → it will sum all pixels under black region and sum of all white pixels under white region → subtract from each other
  - → single value comes out. → like eyebrow

3
- type 3 → apply it to image →  → Finding nose
- we will apply these images to all of region.
- finding the greatest one

Haar

Haar-Features are able to Find Facial landmark Feature (nose, mouth...) by using those kernels.

- viola jones uses 24*24 sub-window to move an image and apply Features.

→ [▯▮] → 2 pixel ⟹ it will apply to all pixels in image → sub-window → value extract

If we increase size of kernel → what happen?

1) tip: increasing the size of each kernel → Features [▯▯▮▮]

Problem: 1) For every single window (24*24) → we have a huge amount of computation → 160,000 Features.

**) position, scale, type → all changes

*) all possible changes in haar-features

Idea: 1) eliminated all redundant features.

- they are not useful
- How to do it? adaboost technique

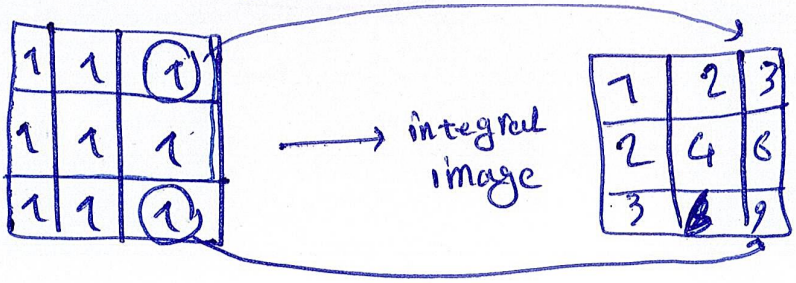
- before diving to adaboost → there is important aspect.

Integral Images: *every single time, we need to sum up all white and black region → Does not look efficient.

- in an integral image the value at pixel (x,y) is sum of Pixels above and to left of (x,y)

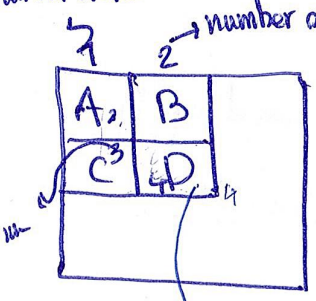
*) every single part I need to sum up all the Pixels in black region and sum up white region

- takes time, computation.
- How to come up



* integral image is a new image matrix value.

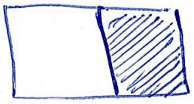
*) when we are going the calculate the value of this Patch



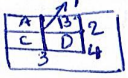
$D = 1 + 4(2+3)$

$D = A + (A+B+C+D) - (A+C+A+B)$

$= D \Rightarrow$



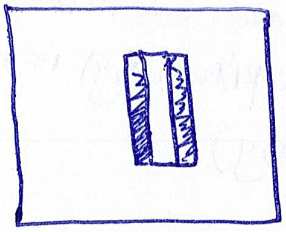
Imagine, we are going to calcelte the value of D Patch in the integral image →



Integral image allows For the calculation of sum of all Pixels inside any given rectangle using only Four values at corner of the rectangle.

- Adaboost technique:
  - eliminate the redundant Feature
  - imagine we hav calculated all computations
    - 160,000 Feature
    - all of them are relevant? no.

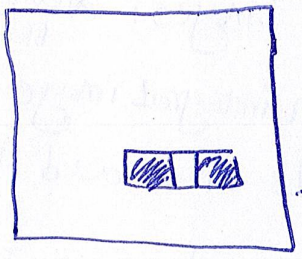
relevant Feature



Image

goal = Pindy nose

gives us great value



Image

goal = Pindy nose

→ we dont have nose here

Irrelevant Feature

low number

which one is relevant and which one is irrelevant?

- it will find certain number of features from all the 160,000 thousand features

Later finds: It will give weight to these features and linear combination of all these feature is used to decide a face or no face

$$F(x) = \alpha_1 \underbrace{f_1(x)}_{\text{weak classifier}} + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \cdots$$

(α_1: 0 or 1; α_2: 0 or 1; α_3: 0 or 1)

strong classifier

weak classifier → relevant feature

output → 1 → great
→ 0 → oh

★) cascading:

every 24 * 24 window → evaluate 2500 features

after performing adaboost and then do the linear combination.

- instead of using 2500 features → use cascade

advantage: applying cascade on any given image we can detect face or not-face.

- Cascade is a group of stages.
- 2500 feature are grouped into several stages
- the job of each image is determining whether a given sub window is definitely not a face. or may be a face.

MTCNN:

- multi task Cascade convolutional neural network
- MTCNN is model tool for face detection
- based on CNN → great performance - robust ★
- contains 3 important layers (levels)

→ 1) Image is resized multiple times to detect face of different size

→ 2) P_net => Proposal network scans → tries to find First_detection → it contain low threshold for detection

→ so it detects so many False positive

خیلی غلط بشی

→ 3) Proposed regions (many False positive) → input for second network (R_net) → refine network

→ tries to find bounding boxes → by filtering detection ★

they are great

→ 4) O_net → apply final refinment → to obtain great and accurate bounding box

Comparision:

Features	viola-jones	mtcnn
speed	very fast	fast
accuracy	good	very good
Robustness	Bad	very good
using color	no	yes
using gpu	—	yes

some advantages of a great detector:

1) Fast

2) accurate → no False positive – no False negatives

3) robust → detects in different size, position – rotation

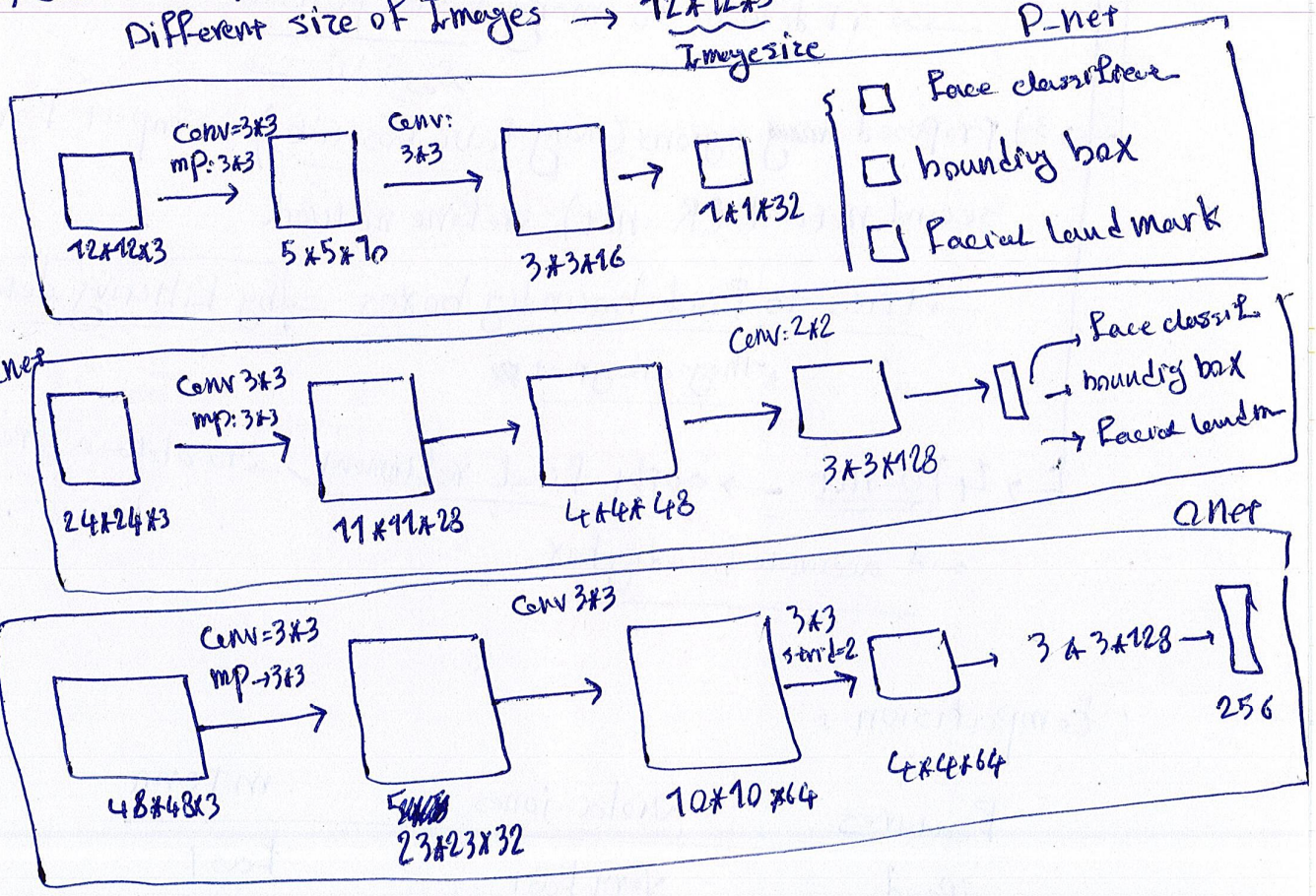
> mTcnn is better than Viola-jones ☆

whole process:

1) Pass in image

2) create multiple scaled copies of the Image → Image Pyramid

Different size of Images → 12*12*3 Imagesize



mtcnn:

- Image pyramid → making a copy of image in different sizes to detect different size of face.

- For each scaled copy → there is 12*12 kernel that will go through every part of the image → aim, searchy for face.
  From (0,0) → (12,12)

- this portion of image is passed → P_net

- P_net gives us the cordinate of bounding box if there is face

- this process will repeat (0+2a, 0+2b) to (12+2a, 12+2b)
  by shifting 2 pixel ⟹

Tip: - small kernel in large image → detect small faces
- large kernel in small image → u large faces

- each network has it's own parameter → P_net network parmeter have been trained → so the output will be a accurate boundybox for each every 12*12 kernel

- network gets a confidence score of each bounding box to make sure about face and delete those that are reliable or low confidence score.

- after collecting all boundy boxes with High confidence score we will have to standerdize → convert to un-scaled image

- non-maximum suppression:
  is method to reduce number of boundy boxes.

*) nms is conducted by first sorting boundybox by their confidence score.

- nms takes the largest boundybox instead of the one the network is most confident.

- calculating the area of each of the kernels.
  overlapping between each kernel and the kernel with highest score.
- delete that has high overlap. → make list of survived kernel.

stage 2) - convertly all boundy boxes to 24*24
- normalize them
- R-net are more similar to Pnet
- gives as near cordinate with high accurate
- again, nms comes in to reduce number of boundy box

> Stage 3:

- before passing bounding boxes From R_net
  - they must convert to 48*48
    similar to previous level → Padding and then Feed
  - these 48*48 size Images go through O_net → cnn base

- outputs of O_net are slightly different From of P_net, R_net.
- It produces three output
  1) cordinate of boundy box → out[0]
  2) cordinate of 5 Facial landmark → out[1]
  3) confidence score → out[2]

- again, we get ride of the boxes with lower confidence.
- standardize both bounding box and Facial land mark cordinate to unscaled image

- another gate: nsm → only one bounding box For each Face

★ short summary of the whole process ★

1) Pass in image
2) create multiple scaled copies of image.
3) Feed scaled image to P_net → cnn base (return bounding boxes)
4) Picking up P_net output
5) Delete bounding boxes with low confidence
6) convert 12*12 Kernel cordinates to unscaled.
7) non-maximum supression → reduces the number of boundy box
8) convert bounding boxes cordinates to unscaled.
9) reshape boundy boxes To square.

Stage 2:

1) Pad-out bound boxes → 24*24

2) Feed scaled image into R-net.

3) gather R-net output.

4) delete boundy boxes with low confidence.

5) nms → reduce number of boundy boxes.

6) convert them to unscaled image cordinate

7) reshape boundy boxes to square

Stage 3:

1) Pad-out bound boxes → 48* 48

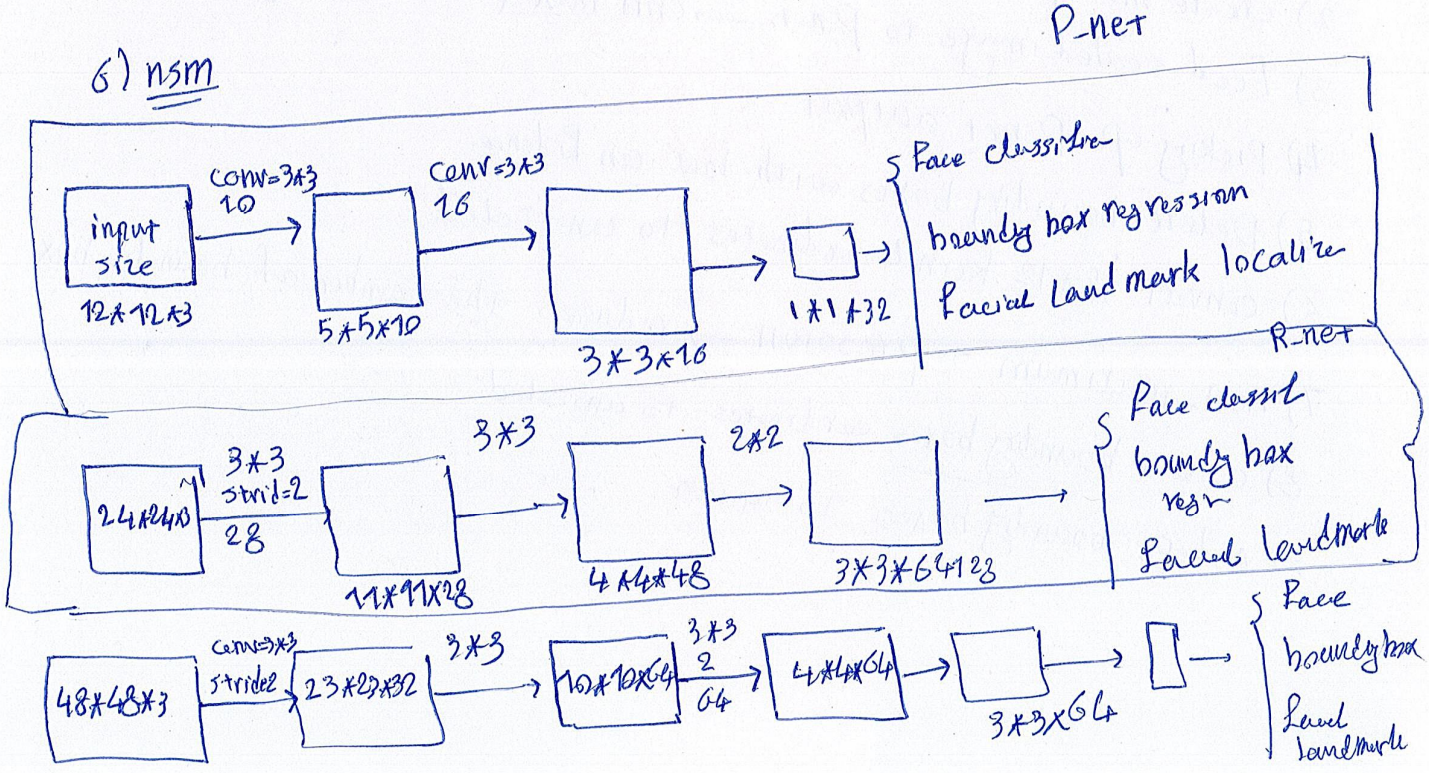
2) Feed scaled images into O-net

3) gather O_net output

4) ~~convert them~~ Delete boundy boxes and ~~bounding boxes~~ with lower confidence

5) convert them to unscaled

6) nsm



- For every image we pass in

  1) network creates Image pyramid → multiple copies of that image in different size.

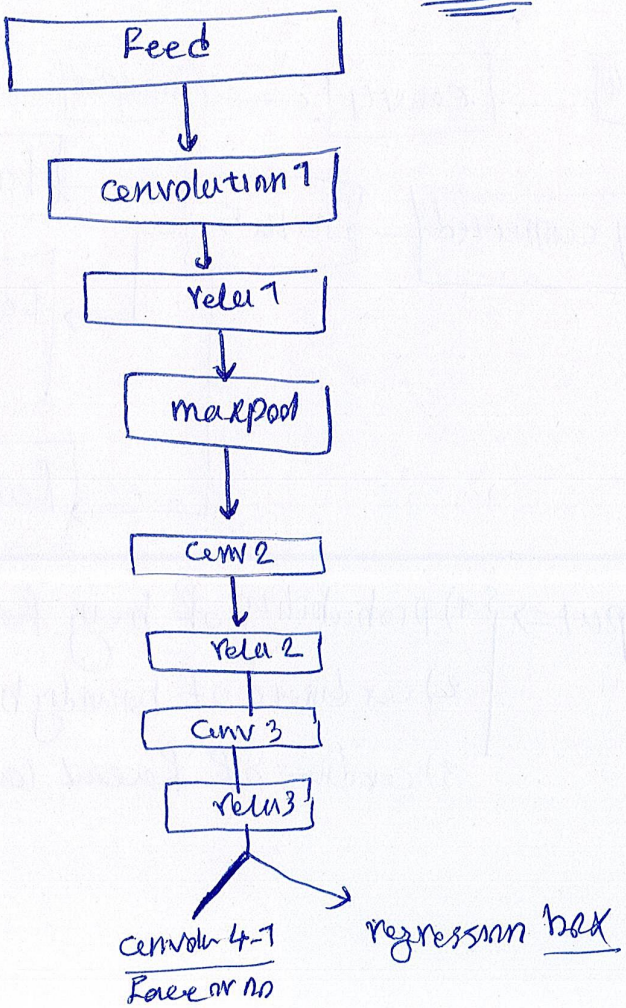
- in P_net, For each scaled image, 12*12 kernel runs through the image searching for Face.

- within each of these 12*12 kernels, 3 convolutions are run with 3*3

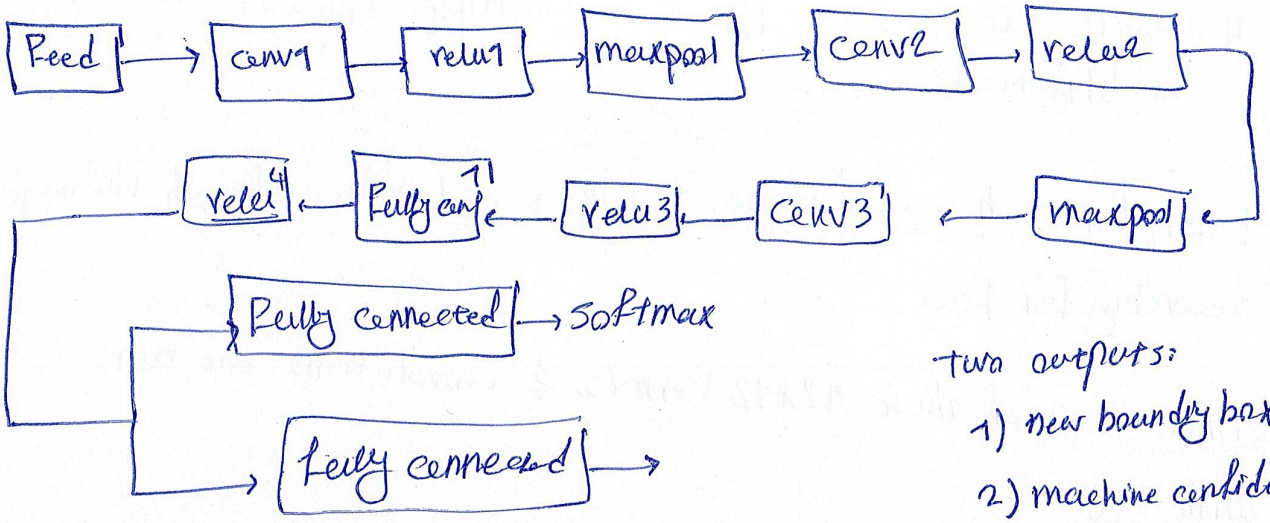
- after every convolution layer → a relu layer → as activation

- then maxpooling → reducing the size of input and take the largest.

P_net

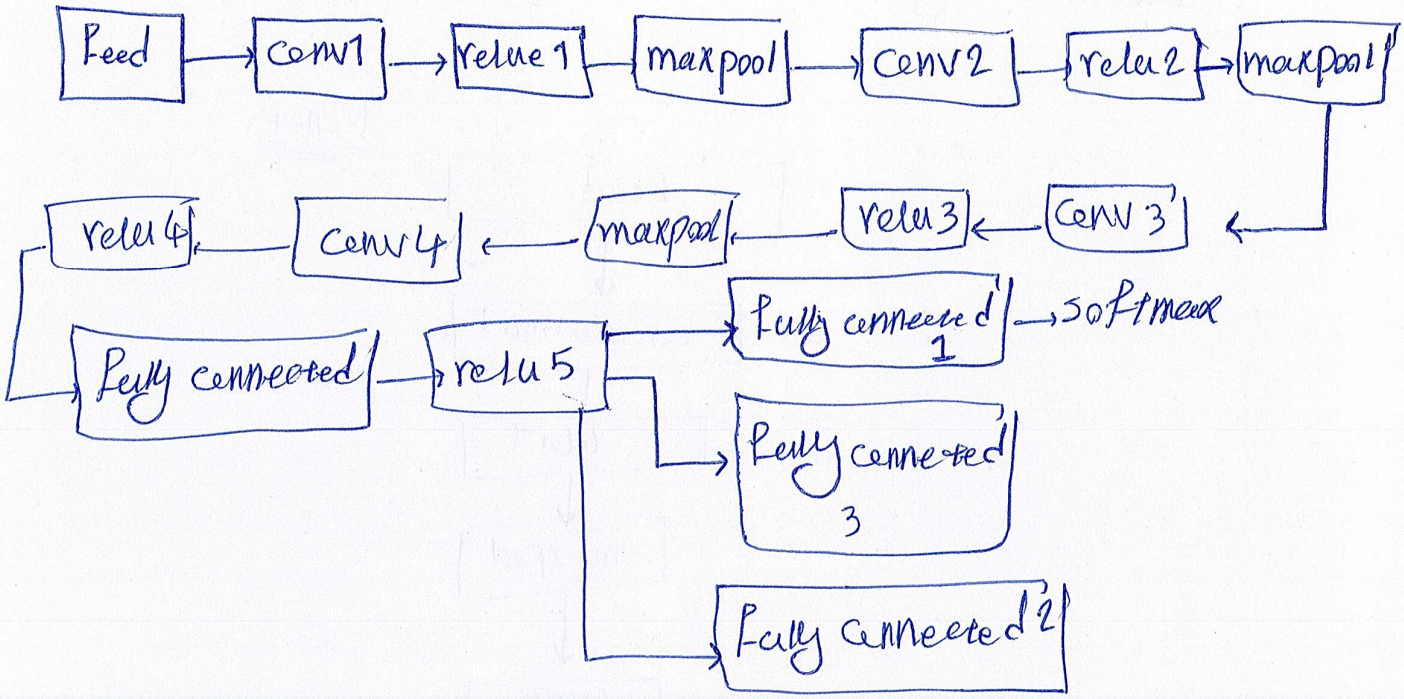


K-net



two outputs:
1) new boundry box
2) machine confidence

O-net



3 output => { 1) Probability of being Face
2) cordinate of boundry box
3) cordin of Faceial landmark

Classification algorithm
- classical approach
- neural network base approach → nnb

- classical approaches → Do not use neural network → most of them around 99% were released before 2012.

- neural network base : using neural network as core → cnn

** classical approaches:
- For many years, they were the best option For classify Images, ...
- with development of computer capability (cpu, Ram, gpu, ...) and new structure, they became weak → new methods took place For some rea
  → 1) complexity / 2) great result
- svm : a Famous, interesting and Flagship when it comes to talk about classification ⇒ 40%

Dynamic Bayesian network:

a summary information:
svm : 40% , Dynamic Bayesian : 10% , Fuzzy : 6,7%
marker model : 5% , knn : 5 , naive bias ⇒ 5 %, ...
Decision Tree : 5% , adaboost : 5%, ...

Some important Dataset for facial emotion recognition

1) AffectNet
- more than one million images
- 1250 emotion → tags
- Diffrent language

2) Fer_2013
- 28,000 labelled images → trainig-st
- 3,500 " " → validation
- 3,500 " " → test set
- Collected by Google Image
- 7 emotion tag →
  - happy
  - sad
  - angry
  - afraid
  - suprise
  - disgu

3) JAFFE
- Japanese Female Facial expression
- 1998
- around 10 Japanese women took picture of herself
- happines, sadness, suprise, anger, disgust, neutral

4) MMI
- create from 25 people of different regions. (Asia, European, ...)
- 44% woman, 55% men
- has many-non basic feature → more than 6 labels.
- each image is 720 * 576

.....

1) SVM approaches:

1) the most famous one - most used one, the most reliable one
2) SVM → is supervised learning.
3) capable of generating input-output mapping function.
   From a set of labeled data with features → produce function
4) classify element by determining boundries (hyperplane)
   seprate class
5) 1963 was developed

2) Dynamic Bayesian network:

3) neural network base approches:

3.1) the use of cnn to solve computer vision has been successful in many areas.

3.2) neural network base → mlp
→ NN
→ cnn
→ summary → for computer vision task: cnn: 66%
neural netw... 21,2
mlp: 12,1%

3,3) Ann (Artificial neural network):

- they all attemp to simulate what happen in human brain.
- they contain set of processing unit → node
- all nods are interconnected with nodes in next layer → via connection → weight (synapth)

*) mlp and cnn are a specific configuration.

3,4) mlp : multi layer perceptron

- using some layer of node to improve degree of accuracy and extract complex Pattern and Features.
- hidden layer → great tools.
- sigmoid and relu function are used to Perform non-linearity ★

3,5) Convolutional neural network : ♡

★ they are great to deal with Image/video

★ take Image/video direcly → Doesn't need to flat you image

★ reduce time

★ number of parameter decreased sharply

★ cnns have 2 important section { 1. Feature extraction / 2. Fully connected

- Feature extraction : using convolvidea to extract feature
- Fully connected → mlp to classify feature map.

Feature selection:

- is one of most important stages for extracting important and relevent feature to have great prediction.

- there are several algorithms for feature extraction in traditional machine learning ((Images))
  1) Active shape model (ASM)
  2) local Binary Patterns (LBP)
  3) HOG => Histogram of oriented gradient

1) ASM: - It was proposed in 1995
- with the goal of generating points that fit the contour of the object.

2) LBP, HOG:
- work by splitting the Image into small pieces called cells and generating histogram.

LBP: provides a comparision between each cell's pixels and their eight neighbors to build a binary number.

Hog: instead of making comparision between pixels, makes a histogram from the gradients.

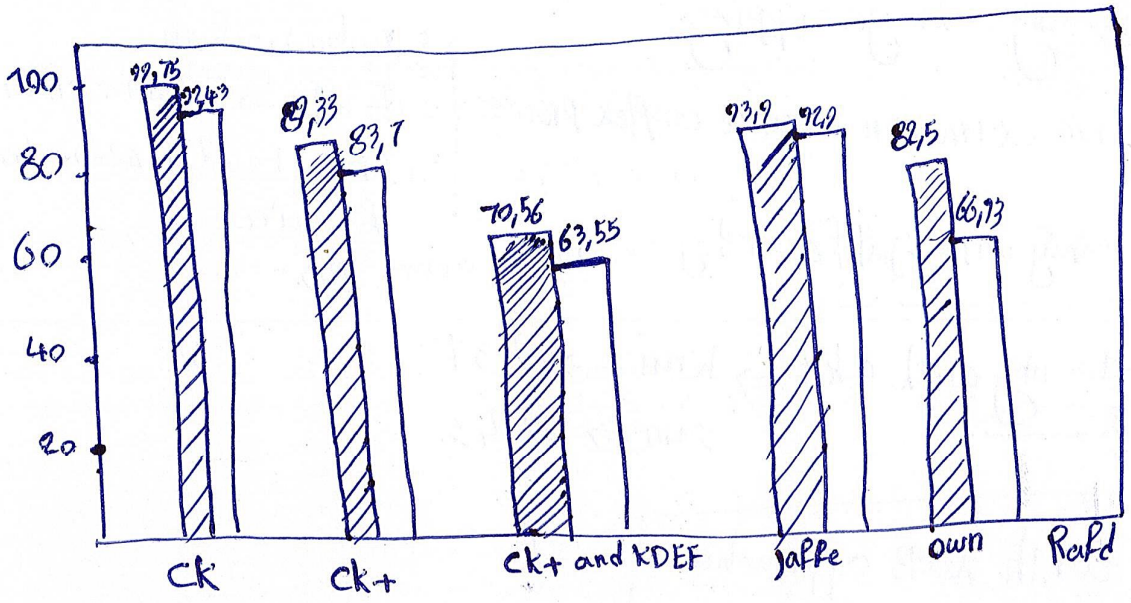
Feature extraction (CNN):

- CNN Does it in First section (Conv layer)
- Conv layer consists of
  - Conv
  - Pooling
  - Strides
  - Kernel
  - activation Function

→ automatically

After passing convol layer → output is sent to Fully connected to make prediction.

emotion Recognition method:



★) in order to achieve good result with cnn model, training dataset must be huge and large.

★) data augmentation → great tool for generating picture

★) when dataset has less data → cnn; model will overfitted. X

that is why some classical approache got great result. ★★★△

Jaffe with classical approach.

Quite complex XX

Preprocessing : Image cropping.

Feature extraction : quite complex process
1) radon transform
2) EMD → minimize effect noise.
3) IMF : based on relation between feature.

training with Jaffe → they were able to achive 100

testing with ckt ⇒ knn → 99.31
svm ⇒ 99.43.

Jaffe with NNB approaches:

cnn → achived 100%.

there were three datasets → training → huge amount of data

jaffe → 213
KDEF → 4900
SFEW → 700
} training

Viola-jones → detecting

★) huge amount of Imge → stay away from overfitting → 100

testing → 100 → JAFFE

test → KDEF → 80

ck+/BU-3DFE NNB approches:

96,76 → ck

91,89 → BU-3DFE

using data augmentation